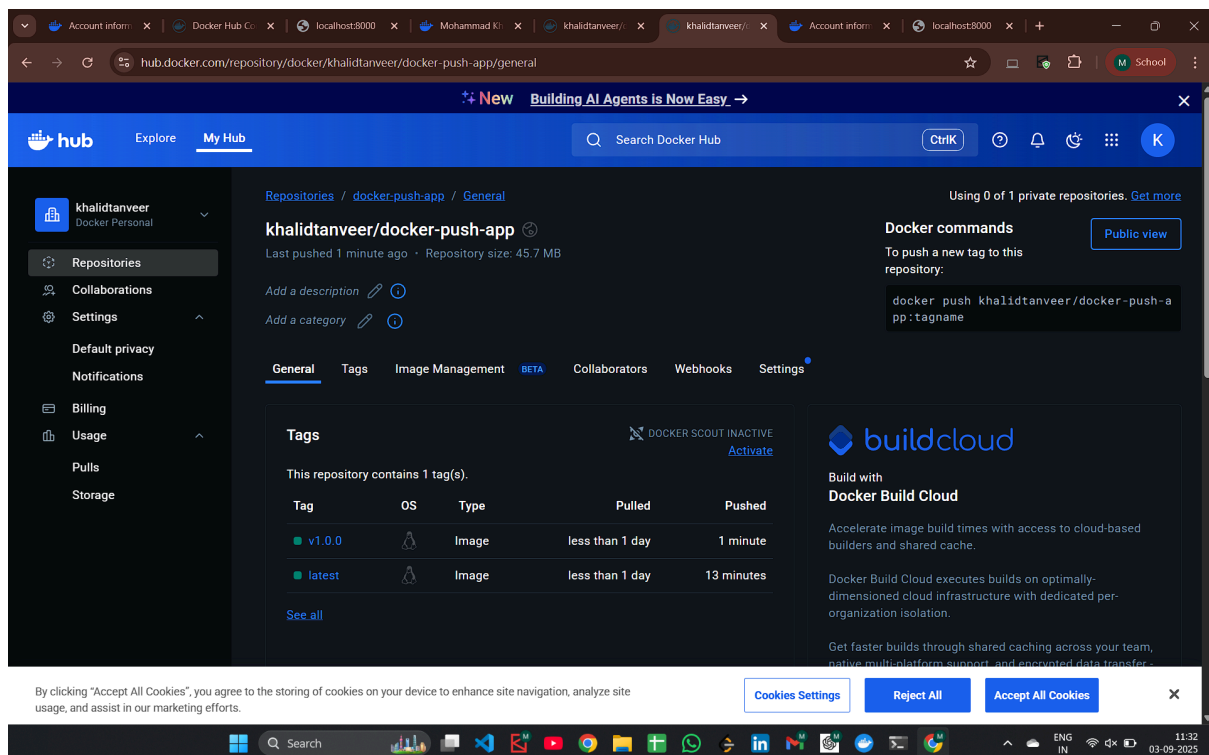


The screenshot shows the Visual Studio Code editor with the `docker-compose.yml` file open. The file defines a service named `app` using the image `khalidtanveer/docker-push-app:v1.0.0`. The service has a build context of `.`, ports `8000:8000`, an environment file `.env`, and a restart policy of `unless-stopped`. A healthcheck is configured to run `curl -f http://localhost:8000` every 30 seconds with a 10-second timeout and 3 retries.

The terminal at the bottom shows the command `docker push khalidtanveer/docker-push-app:v1.0.0` being executed. The output indicates that the image is being pushed to the Docker Hub repository `khalidtanveer/docker-push-app`. The push is successful, and the image is now available on Docker Hub.



## 1. What specific changes or optimizations did you make to your Dockerfile or docker-compose.yml to prepare for deployment?

I optimized the Dockerfile by using `node:18-alpine` for smaller image size, added multi-stage builds, and switched to a non-root user for better security. I also added a `.dockerignore` file to avoid unnecessary files inside the image. In

`docker-compose.yml`, I introduced a restart policy and a healthcheck to ensure production resilience.

**2. What challenges did you face while pushing your image to Docker Hub or running it in a clean environment? How did you overcome them?**

One challenge was Docker rejecting uppercase repository names. I fixed this by renaming the repo in lowercase. Another issue was authentication when pushing; I solved it by generating and using a Docker Hub Personal Access Token instead of my password. Running in a clean environment taught me to prune old images and test with fresh pulls.

**3. Reflect on your learning so far: What part of Docker now feels intuitive to you, and what would you still like to explore or understand better?**

Building and running images locally, tagging, and pushing to Docker Hub feels intuitive now. I understand the flow from `Dockerfile` → `docker build` → `docker run` → `docker push`. I'd like to explore Docker networking, multi-container orchestration with Compose in more depth, and eventually Kubernetes for scaling.